MANAGE RELATED CASES MODULE

1   TECHNICAL REQUIREMENTS

1.1   Introduction

This document is the technical response to the Decompanionator User Requirements document. It will describe how the development team will implement the changes and additions to SWSS Childrens to effect the requirements.

This document is also to be used as a tool by the development team when coding the solution or maintaining it in the future. Thus this document is likely to be updated during the lifecycle of the SWSS project. Versions of this document will be maintained in PVCS, and the reader should be aware that multiple printed versions may exist.

1.2   Module Description

Companion groups are an important part of SWSS foster care cases. However, the groups must be separated when individuals in the group go into adoption at different times, or if they leave the foster care program for any reason. The Decompanionator is a utility to separate companion groups, creating copies of the other members in the group so that no data is lost.

The user needs to select a case from a list of their active cases. Because a user may have several load numbers, a load number must be selected first. Supervisors will have the option of choosing the load number assigned to any user who they supervise.

The companion group for a selected case is displayed. The user may then select companions from the group and split those selected into a new group. The new group may be a single child or several from the previous group. The only restriction is that one of the original companions must remain in the original group.

Split companions can be reassembled into a single companion group only if all of their cases are unregistered.

This module also allows cases to be linked together.   Linked cases associate any two cases with each other regardless of the assigned worker, case state, or physical location. (Recall that companion groups, by definition, are cases assigned to the same load number.) When a companion group is split, a case link is created between the former companions to reflect their association. This module allows explict links to be made at the workers discretion. There is no functionality associated with linked cases; it is a superficial designation made for cermonial purposes only.

Linked cases can be viewed on the Case Summary Detail screen, where all case links are displayed, including links to MAPPER, FAJ prefill links, and explicit links.

Linked cases are determined by the **case_links** intersection database table so it is not as traumatic to change. The only restriction is that Juvenile Justice cases do not have user-created links. There are several different link types, as follows:

LINKED CASE – A link between two cases created by the user in the Case Registration screen or created when companion cases are separated.

PS XFER – A CPS case transferred to Foster Care.

CFC LINK – A Foster Care, Juvenile Justice, or Adoption case was prefilled from an existing, but closed, Foster Care, Juvenile Justice, or Adoption case on SWSS.

DUAL LEGAL STATUS – A Foster Care or Juvenile Justice case was created by SWSS when a Juvenile Justice or Foster Care case was closed using the close codes: "Open to Neglect" and "Open to Delinquency"

1.3   Requirements

1.3.1   Process Description

When a companion group of children's cases no longer function as a group, the group needs to be separated. This happens when one child goes into adoption, or is returned to the family, move into the supervision of a different caseworker, or otherwise leaves FAJ while the remaining companions do not.   The concept of a companion group has more to do with the electronic data regards childrens cases than it does about the childrens cases. Being in a companion group allows case members, legal information, and placement information to be shared between the members of the group. Although this mimics the manner in which the case worker supervises the case there is no formal arrangement of a "companion group" with the foster care, adoption, or protective services programs.

The electronic bond between the members of the group is dissolved and copies of the information are made at the time of the dissolution so that nothing is lost. However, the members then go there own way creating data that can no longer be shared.

A worker may want to link cases to each other to remind them of some relationship between the cases. This is entirely up to the discretion of the worker.

1.3.2   Functional Requirements

Decompanionator must remove individuals from a companion group creating copies of formerly shared information.

1.3.3   Business Events

The following are business events that occur which will initiate use of the features in SWSS Childrens to remove a child from a companion group.
- A child's case closes (and other companions do not)
- A child is adopted (and the others are not)
- A child moves to another county and is assigned to a different load number (and the others do not)
- A worker decides a link to another case is appropriate and needed.

1.3.4   List of Program Units

1.3.4.1   Hierarchical List

The following is a thorough listing of the source code subroutines and shows which unit calls another, including stored procedures on the Oracle Server. Note: PL/SQL stored procedures are in shaded cells.

FIRST CALL DEPTH	SECOND CALL DEPTH	THIRD CALL DEPTH	FOURTH CALL DEPTH
Select.frm			
Form_Load	PreviousInstance Me		

FIRST CALL DEPTH	SECOND CALL DEPTH	THIRD CALL DEPTH	FOURTH CALL DEPTH
	ExtractINI_Info		
	SetDeviceIndependentWindow		
	dbPopPrimeWorker	casereg.GetPrimWorkChoices	
CboAwkr_Click	casereg.loadno_active_cases		
CmdContinue_Click	frmPickComp.Show		
CmdLink_Click	frmLinkCase.Show		
PickComp.frm			
Form_Load	SetDeviceIndependentWindow		
	dbLoadSibs	casereg.getsibstuff	
CmdRemove_Click			
CmdCopy_Click			
CmdContinue_Click	fCreateNewSibGroup	fCallDecompanionate	casereg.decompanionate
		fCallAddLink	casereg.addlink
FrmLinkCase.frm			
Form_Load	tShortCaseName		
	fGetProgram	casereg.get_ProgCode	
	DbPopCFCADPTWorkers	casereg.GetStateProgWorkers	
	PopLinksGrid	casesumm.links	
CboAwkr_Click	casereg.loadno_active_cases		
CmdAdd_Click	FCallAddLink	casereg.addlink	
CmdDelete_Click	FCallDeleteLink	casereg.DeleteLink	

1.3.5 Report (output) Images
Not applicable.

1.3.6 Data Elements
The following table represents the data elements entered on Manage Related Cases.

Form Field Description	Data Source	Data Target	Targe
log id	swss_case.log_id		
case name	swss_case.case_name		
legal status	swss_case.latest_legal_status		
load number	load_worker.load_no		
worker name	worker.worker_name		
case state	swss_case_state.description		

Form Field Description	Data Source	Data Target	Targe
program code	swss_case.program_code		

1.3.7 Integration with Existing System

The Decompanionator (screen name: "Manage Related Cases") is available via the Utilities menu only.

1.3.8 Module Dependencies

This module has no dependencies.

1.3.9 Database Subject Area

The following are the stored procedure interface definitions for interacting with the database:

```
procedure GetPrimWorkChoices
(
      i_worker_id in number    --user worker id
,     i_program   in varchar2 --program where we are adding
,   o_RefCur     in out t_AnyRefCur
);

TYPE LoadNo_Act_Cases_Rec IS RECORD
(
t_log         NUMBER,
t_county_no     county.county_no%TYPE,
t_CASE_NAME     VARCHAR2(90),
t_LEGAL_STATUS  NUMBER(2),
t_SIB_GRP       NUMBER,
t_CASE_NUMBER   varchar2(9)
);
TYPE LoadNo_Act_Cases_Cursor IS REF CURSOR RETURN LoadNo_Act_Cases_Rec;
procedure loadno_active_cases
(
      i_load_no   in    number
,     i_prog_code in    varchar2
,     i_case_state    in    number
,     LoadNoActCasesCursor IN out CASEREG.LoadNo_Act_Cases_Cursor
);


TYPE SibStuff_Rec IS RECORD
(
case_name     VARCHAR2(90),
legal_status  NUMBER(2),
child_id      NUMBER,
log_id        NUMBER,
case_state    VARCHAR2(100),
program_code  VARCHAR2(10)
);
TYPE SibStuff_Cursor IS REF CURSOR RETURN SibStuff_Rec;
Procedure GetSibStuff(
      i_County    in    number     --the county for this log
,     i_SibGrpID in     number            --the sib group number to resurrect
,     GetSibStuffCursor IN out     CASEREG.SibStuff_Cursor
```

```
) ;

Procedure Decompanionate
 (    i_logno in   number
 ,    i_county in number
 ,    i_proposed_sib_grp_id in number
 ,    o_new_sib_grp_id out number
 ,    o_sqlcode out number
 ,    o_sqlmessg out varchar2
);
Function AddLink
 (     pSourceCounty      in     number
 ,     pSourceLog   in    number
 ,     pDestCounty in     number
 ,     pDestLog    in     number
) return number;

Function DeleteLink
 (     pSourceCounty      in     number
 ,     pSourceLog   in    number
 ,     pDestCounty in     number
 ,     pDestLog    in     number
) return number;

function get_ProgCode
 (
       i_legal     in     number
)
       return varchar2;

TYPE Load_No_Rec IS RECORD
(   load_no load.load_no%TYPE
,   worker_name worker.worker_name%type);
TYPE Load_Nos_Cursor IS REF CURSOR RETURN Load_No_Rec;
--
PROCEDURE GetStateProgWorkers
(
       i_program             in load_program.program_code%type
,      GetStateProgWrksCursor IN OUT CASEREG.Load_Nos_Cursor
);

Procedure Links
--
--Return a recordset of all the linked cases, where links may refer to
--PS Xfer source cases, CFC prefills, and Linked cases
--This mostly looks for associations made in the CASE_LINK table
--
(      i_logno     in     number
,      i_county in number
,    io_RefCur in out t_AnyRefCur
);
```

1.3.10 Data Warehouse

Not known.

1.3.11 Technical Issues

None.

1.3.12 Test Plans

Verify the following:

- The newly created companion group must have a copy of all the data regarding any members of the former companion group that are not included in the new group.
- The former companion group must have a copy of all the data regarding any members of the newly created companion group.
- If the user performs an action upon the database after the database has timed the user out, SWSS must automatically reconnect to the database and continue working.
- Central Office users (county 84) need inquiry access for case information.
- The changes made on the Link Case screen must be reflected on the Case Summar— Detail screen.